electronics

MDPI

Article

Optimization of the AODV-Based Packet Forwarding Mechanism for BLE Mesh Networks

Muhammad Rizwan Ghori [1], Tat-Chee Wan [1,2,*], Gian Chand Sodhy [1] and Amna Rizwan [3]

[1] School of Computer Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia; mrizwanghori@student.usm.my (M.R.G.); sodhy@usm.my (G.C.S.)
[2] National Advanced IPv6 Centre, Universiti Sains Malaysia, Penang 11800, Malaysia
[3] USM Family Accommodation, Penang 11800, Malaysia; amnarizwan87@outlook.com
[*] Correspondence: tcwan@usm.my; Tel.: +60-4-653-3004

Abstract: The standard Bluetooth Low-Energy mesh networks assume the use of flooding for multihop communications. The flooding approach causes network overheads and delays due to continuous message broadcasting in the absence of a routing mechanism. Among the routing protocols, AODV is one of the most popular and robust routing protocol for wireless ad hoc networks. In this paper, we optimized the AODV protocol for Bluetooth Low-Energy communication to make it more efficient in comparison to the mesh protocol. With the proposed protocol (Optimized AODV (O-AODV)), we were able to achieve lower overheads, end-to-end delay, and average per-hop one-way delay in comparison to the BLE mesh (flooding) protocol and AODV protocol for all three scenarios (linear topology with ten nodes, multipath topology with six and ten nodes). In addition, the proposed protocol exhibited practically constant route requests and route reply setup times. Furthermore, the proposed protocol demonstrated a better Packet Delivery Ratio (PDR) for O-AODV (84%) in comparison to AODV (71%), but lower than the PDR of the mesh (flooding) protocol with 93%.

Keywords: Bluetooth Low-Energy; BLE mesh networks; AODV

check for updates

Citation: Ghori, M.R.; Wan, T.-C.; Sodhy, G.C.; Rizwan, A. Optimization of the AODV-Based Packet Forwarding Mechanism for BLE Mesh Networks. *Electronics* **2021**, *10*, 2274. https://doi.org/10.3390/electronics10182274

Academic Editor: Stefano Rinaldi

Received: 19 August 2021
Accepted: 14 September 2021
Published: 16 September 2021

1. Introduction

Bluetooth Low-Energy (BLE) is a Wireless Ad Hoc Network (WAHN) technology that is becoming increasingly popular among IoT devices that run on batteries. The Bluetooth Special Interest Group (SIG) introduced the BLE standard in Bluetooth Version 4.0, which was further improved in Bluetooth Versions 4.2 and 5 [1]. For multihop communications and network connections, BLE 4.x initially used the traditional Bluetooth-based Personal Area Network (PAN) paradigm. BLE 5 aims to address these flaws by implementing a flooding-based mesh architecture, which will enable better coverage of the network, standardized intercluster communications, and improved security [2]. The model, foundation model, and access, upper/lower transport, network, and bearer layers make up the BLE mesh system architecture, which sits on top of the BLE network stack [3].

Despite its many features, the BLE mesh protocol assumes the use of flooding for multihop communications. The flooding approach results in network overheads and delays due to continuous message broadcasting in the absence of a proper routing mechanism.

Aside from the flooding-based protocol, numerous common routing protocols for wireless ad hoc networks exist, such as AODV [4], DSR [5], and others. AODV has been used in a wide range of applications and scenarios, demonstrating its dependability and robustness.

For this paper, we considered a multihop mesh with a flat/nonhierarchical topology as a usage scenario and proposed a O-AODV protocol for BLE by replacing the network layer protocol in the BLE mesh architecture to improve communication efficiency and reliability over the current AODV and mesh protocols. The objectives of this research were to: (a) optimize BLE mesh performance by reducing overheads and delays in comparison with the flooding approach and (b) improve the Packet Delivery Ratio (PDR) performance of AODV.

Except for the work in [6] (the author tested with a single topology with six nodes (two hops)), there is no significant research available for improving the BLE mesh protocol. The AODV protocol was demonstrated to be the best among other ad hoc network protocols [7,8]. As a result, as discussed in Sections 4.1.1, 5.1 and 5.8, we optimized the performance of the current version of the BLE-based AODV protocol for this article by reducing mainly the packets' retransmissions to reduce the delays, overheads, and traffic load. Furthermore, we conducted experiments with three topologies (linear—nine hops from the source to the sink with ten nodes, multipath—two hops from the source to the sink with multiple path availability with six nodes, partial mesh—four hops from the source to the sink with several paths with ten nodes) to evaluate the performance of our proposed protocol to that of the existing protocol.

In addition, before moving on to the other sections of this paper, it is required to examine the application scenario of a BLE-based mesh network in order to increase the readability. Wireless Ad Hoc Networks (WAHNs) are gaining popularity as they require no infrastructure and low-power communication. When it comes to hospitals, infrastructure-less communication is required to deal with emergency situations, such as calling the staff in an emergency or normal condition from one location to another, message transfer, and patient convenience to obtain various types of information after entering the hospital's vicinity. Many technologies are available to make WAHN practical for hospitals, including ZigBee, Z-Wave, threads, BLE, and others. In light of the foregoing and following extensive research, Bluetooth Low-Energy 5 (BLE 5) is the most suitable technology for this kind of network because of its easy availability in mobile devices, low power, low cost, and mesh support characteristics. Due to the critical nature of the hospital, there is a requirement for a very efficient and robust BLE-based mesh communication network as the hospital is dealing with emergencies all the time. Consequently, there is a strong requirement for a suitable mesh protocol that can make hospital communication effective and robust. Therefore, as per the literature review, during the last few years, researchers have presented different strategies (flooding-based and routing-based) to enable the mesh topology for BLE. Flooding is a simple solution because it does not involve the formation of a connection or the routing protocol [9]. Due to the lack of a routing protocol, it avoids route formation delays and consumes less memory. However, flooding is inefficient because there would be a flood of messages for successful end-to-end communication, which can cause network congestion and overheads and, hence, consumes more power resources. To address the concerns, the routing strategy is best suited for large networks that require efficiency. According to the literature review, researchers have made certain recommendations for the implementation of the BLE mesh routing protocol. Furthermore, regarding the routing-based mesh solution, a single BLE-based protocol is available from [6] presenting an AODV-based solution for BLE mesh networks. However, there are still issues that have to be resolved as stated in the aforementioned research objectives. Moreover, we concentrated on stationary IoT devices for the hospital situation, as well as for O-AODV protocol experimentation.

The organization of the rest of the paper is as follows: Section 2 discusses the BLE mesh architecture and message routing protocols, while Section 3 discusses related works based on the BLE mesh. Section 4 talks about the Optimized-AODV (O-AODV) protocol topologies and testbed design. Section 5 is devoted to the proposed optimized-AODV protocol optimizations achieved with the experimental results, the discussion, and research contributions, while Section 6 concludes the discussions for the paper.

2. BLE Mesh Architecture and Message Routing Protocols

To improve the readability of the paper, it is necessary to discuss the background material in the following sections, such as the BLE mesh architecture and the available mesh routing protocols, before delving into the details of the proposed protocol.

2.1. BLE Mesh Architecture

To describe various words and concepts used throughout the paper, a quick introduction to the BLE mesh layers is presented and shown in Figure 1:

- Model layer:
  Models define operations based on use cases in the model layer. The Bluetooth mesh model specification or vendors specify these models. The models are identifiable by Bluetooth SIG and vendor-defined unique identifiers of 16 and 32 bit;
- Foundation layer:
  The states, messages, and models are defined in this layer that are needed to setup and maintain mesh networks in different contexts. The configuration client and server model and the health client and server model are the two types of models specified in the Bluetooth SIG specification;
- Access layer:
  The access layer specifies how the upper transport layer can be used by the upper layers. It is also responsible for defining the app's data structure and implementing encryption/decryption operations. Finally, before passing data to the upper layers, it ensures that the network and application keys in the incoming data are correct;
- Upper transport layer:
  This layer is in charge of application data encryption, decryption, and authentication, as well as keeping the secrecy of access messages. It also specifies the control messages that are used to coordinate the transport layer functions between nodes;
- Lower transport layer:
  This layer specifies how messages in the top layers are segmented and reassembled in the lower-layer protocol into units of data. It is also in charge of controlling segmentation and reassembly;
- Network layer:
  Data transfer addressing, formatting, encryption, and authentication are all handled by the network layer. This layer is also in charge of handling message forwarding and dropping decisions.
  The BLE mesh network layer, on the other hand, lacks a routing mechanism and relies on a flooding-based approach;
- Bearer layer:
  The message transmission mechanism is defined by the bearer layer. In the newest BLE 5 mesh requirements, there are two types of bearers available: advertising bearer and GATT bearer;
- Bluetooth core specifications:
  The Bluetooth core specifications' network stack supports the physical data transfer between the nodes/devices with the help of a layering architecture. The host, controller, and physical/radio layers are the three fundamental layers of the BLE network stack.
  The host layer, which sits just beneath the application layer, has many nonreal-time network and transport protocols that allow apps on different devices to communicate with one another. The Generic Access Profile (GAP), Generic Attribute Profile (GATT), Security Manager (SM), Attribute Protocol (ATT), and Logical Link Control and Adaptation Protocol (L2CAP) are among the modules found in this layer.
  The BLE Link Layer (LL) protocols (low-level and real-time) are implemented by the controller layer. It performs packet reception, schedules transmissions, and ensures data delivery to the destination, in addition to handling control operations and physical layer interfaces via the Host Control Interface (HCI);
  The physical layer is in charge of wireless signal transmission. BLE uses the 2.4 GHz Industrial, Scientific, and Medical (ISM) frequency spectrum, with 40 narrowband channels (2 MHz bandwidth) divided into three Advertising Channels (AC) (Ch. 37–39) and 37 Data Channels (DC) (Ch. 0–36). Device detection, connection establishment, and the transmission of broadcast messages are all handled by the ACs. DCs, on the other hand, allow two-way data flow between linked devices and rely on Adaptive

Frequency Hopping (AFH) for subsequent communications.
Moreover, the BLE communication profiles are discussed in Appendix A.

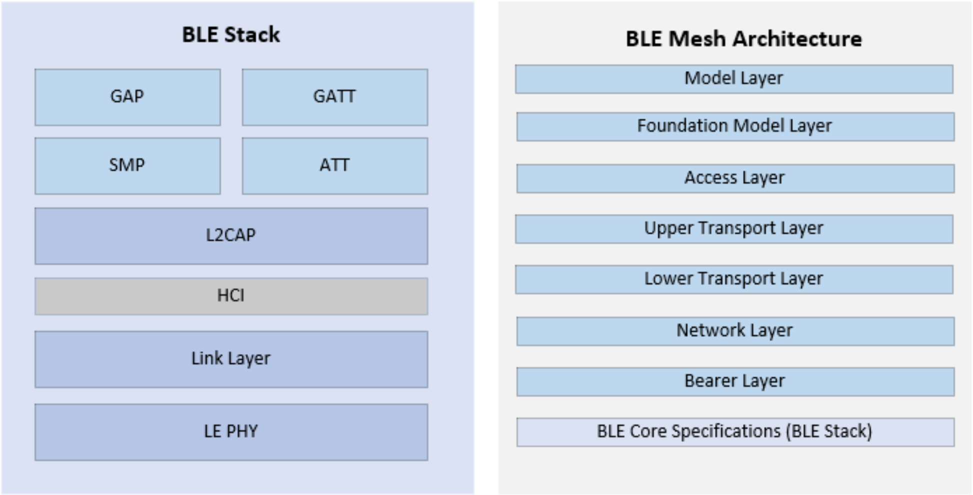

Figure 1. BLE stack and mesh architecture [3].

2.2. Message Routing Protocols

The BLE mesh protocol relies on the flooding approach that causes network overheads and delays because of continuous broadcasting. Therefore, this section gives a brief overview of three popular routing-based messaging paradigms adopted by different multihop transmission protocols to enable readers to better understand the implementation of the proposed O-AODV protocol:

- Reactive (on-demand) protocols
  Messages received by the reactive forwarding protocol provide information about the destination nodes. Every transmission table entry is only active for a limited amount of time. The item will be erased if no traffic is encountered for a certain destination within the specified time frame. A process for new route discovery will be launched upon receiving the request from the sender node. Examples of such protocols include ad hoc AODV [4] and Dynamic Source control Routing (DSR) [5];
- Proactive (table driven) protocols
  For all nodes, whether active or not, explicit forwarding table entries are maintained by proactive forwarding protocols. The Bellman–Ford algorithm is used to keep feasible pathways to important nodes, allowing data to be delivered to the destination as soon as possible. The examples of this kind of protocol include: Babel, Optimized Link State Routing (OLSR) [10], Destination Sequenced Distance Vector (DSDV) [11], and others;
- Cluster-based protocols
  Scatternet is a kind of cluster-based transmission protocol in BLE 4.1 to support multihop communications. The network nodes are divided into multiple overlapping disjoint clusters [12]. Each cluster is led by the cluster head, who maintains memberships in a cluster and is then utilized to find the path that connects the clusters. The clustering of nodes decreases the flood during the path discovery procedure. In addition, the protocol monitors any single directional links for transmission between clusters and intracluster. Protocols of this type include the Two-Tier Data Dissemination Protocol (TTDD) [13], ring routing [14], Energy-Efficient Secured Ring Routing (E2SR2) [15], and others.

In a BLE mesh network with identical battery-operated nodes, a flat topology would provide the most flexibility to support multihop communications among the nodes. Reactive protocols such as AODV compute routes on-demand, eliminating the need for periodic updates [4] and reducing message flooding. Furthermore, because BLE nodes are typically resource constrained, the reactive protocol is best suited because it requires less storage than proactive and cluster-based protocols.

3. Related Works

In order to identify the research gap, we reviewed the latest BLE-based mesh communication protocols.

The early BLE mesh solution proposed by [16] is called MultiHop Transfer Service (MHTS). The proposed solution also works well for the two-hop and three-hop transmission of data. Reference [17] used the available mesh protocol to construct the mesh topology on office doorbells. According to [18], it is possible to support long-range communication due to the incorporation of the scatternet topology in Bluetooth 4.1. The author introduced the concept of service mediation based on Name Data Networking (NDN). In addition, BLE multihop routing was developed by [19], based on a scatternet topology feature of the BLE 4.1 protocol. To increase production, Reference [20] developed a BLE-based data collection approach for the agriculture area. Moreover, Reference [21] proposed the design for a multihop tree-based wireless network implementation, Furthermore, Reference [22] proposed a dynamic node organization approach for data routing, through an on-demand routing technique. In order to allow mesh functionality in an existing BLE beacon network, Reference [23] presented a novel BLE-based overlay mesh that addresses issues related to the best-effort scheduling and RSSI- based bounded flooding. Likewise, References [9,24] gave performance measures using the existing protocols, i.e., trickle and FruityMesh. Subsequently, a directional ad hoc on-demand multipath distance vector technique was proposed by [25]. By providing a directional link quality indicator for the routing schemes, the protocol overcame the problem of moving nodes' transmission instability. Furthermore, Reference [26] developed a BLE protocol to unravel the weaknesses in the BLE specifications on industrial wireless sensors networks. The main idea behind the article was to divide a network into different subnetworks that were each controlled by a master. To tackle the problem of overlapping events and enable real-time communication, Reference [27] suggested a BLE-based real-time multihop communication network with bounded message delays. Moreover, Reference [28] introduced BLE's multihop on-demand routing protocol, proposing topology configuration and recovery procedures. Reference [29] talked about using BLE technology to automate parking systems. Reference [30] also worked on a specific relay distribution in a flooding-based BLE mesh network with collision mitigation to increase the overall PDR. Moreover, an AODV-routing-based BLE mesh protocol was proposed by [6]. According to the author, less traffic was generated by the protocol than the mesh.

The aforementioned works attempted to improve the BLE mesh protocol features, as well as experimented on the available BLE mesh protocol, but did not focus on improving the underlying protocol. Only [6] has worked on a routing-based protocol to improve the performance of the packet-forwarding mechanism. As a result of the preceding, in this paper, we focused on improving the protocol proposed by [6] by reducing overheads, retransmissions, and delays. Moreover, the selected protocols mentioned in this section are summarized in Table 1.

Table 1. Summary of various pure BLE-based mesh protocols.

Ref.	**–Protocol**	**-Connection-Oriented (C) /Routing (R) -Flooding (F)/(R)**	**Testbed (T)/ Simulation (S)**	**PDR**	**End-to-End Delay**	**Overheads**	**Power Consumption**	**Nodes Other Measurements (OM) Throughput**
[9]	–FruityMesh and Trickle	C (FM)/ R-Reactive F-TR	T-nRF52 (BLE 5) -Five Hardkernel Odroid-C2 -Netgear GS108T 8-port switch	-FM: 100%, 90%, 40% with 1, 5, 10 p/s resp -TR: 100%, 80%, 38% with 1, 5, 10 p/s resp	-FM: 0.3, 3.7, 3.9 s with 1, 5, 10 p/s resp -TR: 0.4, 0.3, 0.3 s with 1, 5, 10 p/s	×	FM: 9 mW TR: 28 mW	37 Nodes
[24]	–FruityMesh	C (FruityMesh (FM)) /R-Reactive	T-Nordic Thingy: 52 IOT sensor kit -nRF52DK -nRF6707	×	×	×	Note: Connection Interval (CI) (7.5–400 ms) (a) 0.65–0.1 mA (Adv Interval 100 ms with CI) (b) 0.6–0.03 mA (Adv Interval 600 ms with CI) (c) Network Life 10–250 d with CI	1 to 3 **Nodes** **OM** with CI from 7.5–400 ms and advertising interval (100 and 600) the current drain is 0–0.65 mA **Throughput** Approximately from 8–0.5 kB/s for CI 5–400 ms and max 3 packets/interval -with CI > 400 ms is 150 b/s
[25]	–D-AOMDV	F	S-MATLAB	Approximately 75–88% with number of nodes 10–40	×	×	×	40 Nodes
[26]	–MRT-BLE	C (Static Routing Configure Offline to obtain bounded delays)	T-(BLE) X-NUCLEO-IDB05A1	1 Hop => 100% 5 Hops => 97%	1 Hop => 120 ms 5 Hops => 1400 ms	×	×	8 Nodes
[27]	–RT-BLE	C (Static Routing Configure Offline to obtain bounded delays)	T-(BLE) X-NUCLEO-IDB05A1	×	20 ms	×	×	4 Nodes
[28]	–CbODRP	C/R-Reactive	S	×	-Route discovery delay 40–100 ms with 50–90 nodes	-Control Packet Overhead 12–88 with Route Discovery interval 1–10 s	Approximately 250–500 mA with 50–90 nodes	50 to 90 **Nodes**

[30]	–K2 Pruning
	Greedy Connect
	and Dominator
		F-(Greedy Connect (GC)
		K2 Pruning (KP)
		Dominator(D))
			S-MATLAB	-Area (330 × 330 msq)
				K2: Approximately 80–8%
				with packets 5–200 p/s
				GC: 65–8%
				with packets 5–200 p/s
					×	×	×	1000 Nodes

[31]	–BLE-Tree Network	C and R (Reactive)	T Raspberry Pi 3 Model B (BLE 4.1) S	100% (for 2 p/s) 97.5% (for 5 p/s) 82% (for 10 p/s)	Round-Trip Time: For 1 to 6 Hops = 100 ms to 530 ms	×	×	40 Nodes
[32]	–BLE Mesh	C/R-Reactive	T- nRF52 (BLE 5)	High	Low	×	×	12 Nodes
[33]	–DC-BMN	C	S-MATLAB	×	×	×	×	100 **Nodes** **OM** For N Slot 10
[6]	–AODV Based BLE Mesh Network	R-Reactive	T- nRF52840DK	69%	6200 ms	42%	×	6 Nodes **OM:** Average RREQ to RREP Setup Time: 6800 ms Traffic Load: Lower Than Mesh and Higher Than This Paper
O-AODV (Ghori, M.R. et al.)	–Optimized AODV Based BLE Mesh Network	R-Reactive	T- nRF52840DK	82%	5300 ms	22%	×	10 Nodes **OM:** Average RREQ to RREP Setup Time: 5800 ms Traffic Load: Lower Than Mesh and by [6]

4. Optimized-AODV Protocol Topologies and Testbed Design

This section describes the topologies and testbed architecture, the hardware and software used, and the discussion related to the proposed optimized-AODV for BLE mesh.

4.1. O-AODV Topologies and Testbed Design

To evaluate the proposed O-AODV protocol in this research, we developed a testbed as depicted in Figure 2 and the topology diagrams in three formats: linear topology with ten nodes as shown in Figure 3, multipath topology with six nodes, and partial mesh multipath topology with ten nodes, as depicted in Figures 4 and 5.

Furthermore, it is important to note that we increased the complexity of the Scenario 3 topology from the six-node topology in order to evaluate the O-AODV protocol's efficiency and robustness. Therefore, we included traffic results for the same four paths for O-AODV and AODV in the Experimental Results Section to demonstrate clearly the protocols' efficiency.

For simulation, Renode software was the only option for simulating the Zephyr RTOS application; it has support for a variety of boards [34]. Although the nRF52840 board is supported by Renode, it currently lacks a radio connectivity feature. Consequently, this paper focused on the experimental evaluation of the proposed protocol, which has the additional advantage of providing insights into various implementation constraints of the protocol in actual devices.

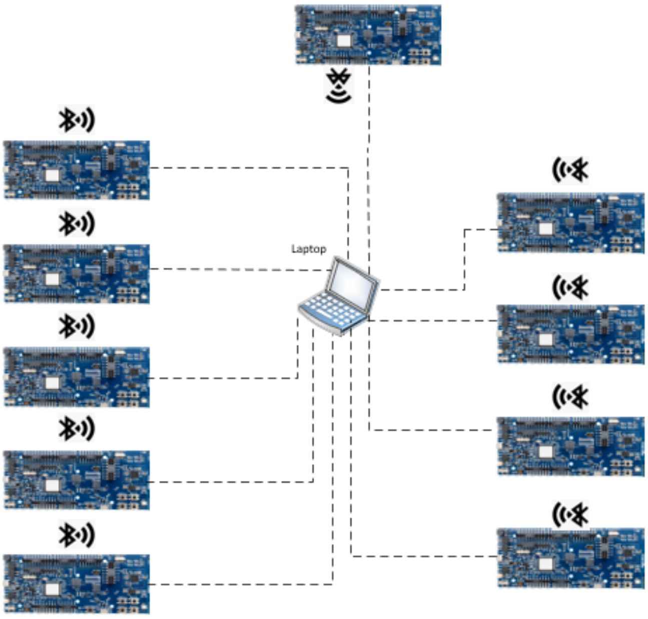

Figure 2. Testbed setup showing the USB links used for the experiment configuration and data collection.

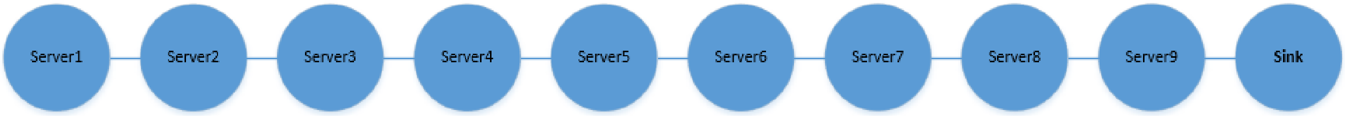

Figure 3. Scenario 1: linear topology with 10 nodes.

4.1.1. Initial Findings and Proposed Optimizations for O-AODV

In comparison to the mesh (flooding) protocol, the original AODV protocol was not performing on par in terms of the packet delivery ratio, overheads, end-to-end one-way delay, and average per-hop delay. With 2–3 nodes, the AODV protocol performed similarly to the mesh protocol, but as the number of nodes increased, the packet delivery ratio declined and the overheads rose, causing real effects on protocol reliability based on initial experiments not reported in this paper. As AODV is a more dependable protocol for ad hoc networks than mesh since it forwards packets based on the route, consequently, we improved the protocol by changing the host (software) part of the BLE stack (which contains the BLE mesh architecture) to improve its performance in terms of packet delivery ratio, overheads, end-to-end one-way delay, and average per-hop delay and to bring it as close to the mesh protocol as possible. We also changed the configuration in the controller component of the stack to enhance the protocol's performance. With the aforementioned improvements, the proposed protocol produced consistent results when compared to the existing AODV protocol, as well as comparable PDR with lower overheads and average per-hop delay and end-to-end one-way delay when compared to the mesh protocol, as discussed in Section 5.7.

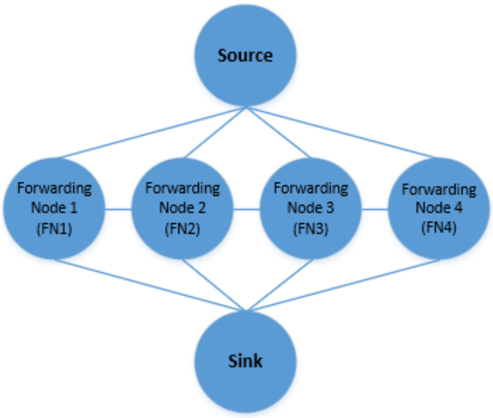

Figure 4. Scenario 2: multipath topology with 6 nodes.

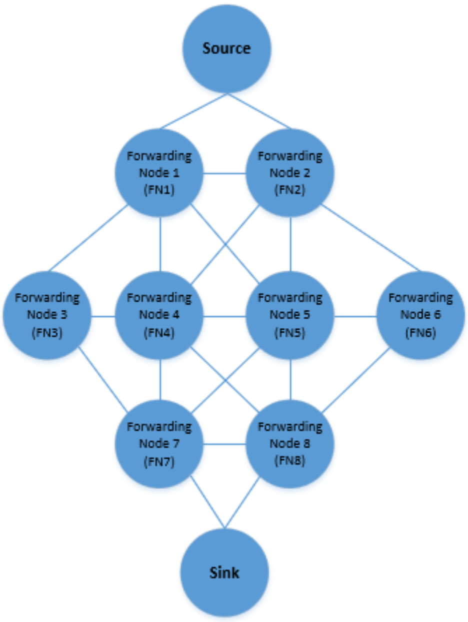

Figure 5. Scenario 3: multipath partial mesh topology with 10 nodes.

4.1.2. Hardware and Software Utilized for the O-AODV Testbed

The Nordic Semiconductor nRF52840 development boards are the BLE nodes selected for the testbed. The boards are embedded with four LEDs and four user-programmable buttons, as well as various input and output interfaces; Segger J-Link OB support; an integrated printed circuit board antenna; a battery (coin-cell) and micro-USB connectivity support used for programming, as well as for powering, and a UART serial port communication. A 32 bit ARM Cortex M4F processor is embedded with 512 kB of flash memory storage, and the board is equipped with 64 kB of RAM.

The only element in the mesh network are the BLE nodes. They were programmed with a BLE stack and the proposed mesh-based protocol that was designed and developed mainly in the application and network layer of the Zephyr RTOS (real-time operating system). All 10 nRF52840 DKs were connected via a USB connection to a laptop for data collection purposes by utilizing the PuTTY software.

The proposed protocol was embedded with the device-provisioning feature for the mesh network security, which means that every device wishing to join the network must first go through the provisioning process. Thus, the built-in provisioning APIs of Zephyr RTOS were utilized to enable the provisioning feature in the proposed protocol's application layer. Furthermore, for physical provisioning, a BLE mesh Android application was required on a smartphone to grant permission to the mesh-enabled nodes to enter the mesh network.

Moreover, the developed protocol was hardcoded with the network, application, and device keys for testing purposes. However, to enhance the security of the protocol, developers could dynamically give the security keys and utilize the available BLE security features as discussed in Appendix B.2.

Moreover, the operating system used to develop the protocol was Zephyr RTOS [35]. Zephyr OS is based on a very small footprint kernel for embedded devices that are resource constrained, which includes anything from simple embedded environmental sensors and LED wristbands to smart IoT apps and comprehensive integrated controllers. More explanation is given in Appendix B.

4.1.3. Experimental Topologies

In this section, we describe the topologies that we used in our experiments. For preliminary testing, we used a simple 10-node linear topology (as shown in Figure 3 by transmitting 100 packets from the source and reduced Tx power up to −40 dB (to make the testbed manageable) to compare the proposed protocol's efficiency to the AODV and mesh (flooding) protocols. Furthermore, to test the proposed protocol's efficacy, we experimented with a multipath simple topology with 6 nodes and with more complex/partial mesh scenario with 10 nodes, as depicted in Figures 4 and 5, with the same configuration as discussed for the linear topology.

5. Proposed Optimized-AODV Protocol Optimizations Achieved with the Experimental Results

In this section, we go over the O-AODV optimizations achieved and the experimental results obtained with those improvements.

Moreover, before delving into the details of the protocol optimizations, it is required to provide an overview of the message transmission flow following the incorporation of the AODV layer.

Figure 6 depicts the BLE mesh communication with the inclusion of the AODV layer, and also shows the layers where the optimizations were performed. In the figure, the message is transmitted from the application layer, which then passes it to the AODV layer for routing. The AODV layer then sends the message to the BLE mesh layers, and the BLE mesh network layer finally hands over the message to the BLE stack, which then sends it to the other device. Furthermore, on the receiving end, the BLE stack sends data to the BLE mesh layers, which ultimately send it to the AODV and application layer.

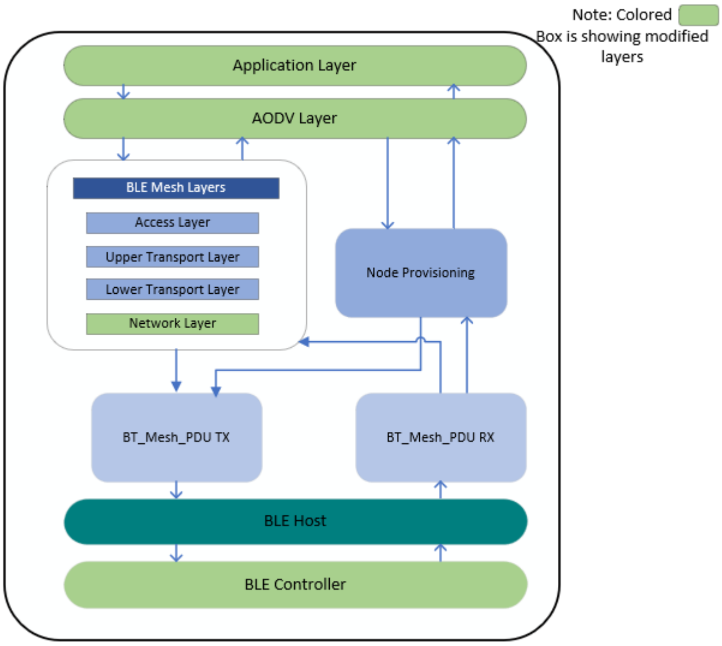

Figure 6. Message passing diagram with the AODV layer.

5.1. How the Optimizations Were Achieved

This section discusses how we improved the proposed O-AODV protocol's efficiency in comparison to its predecessor version.

5.1.1. Route Request and Route Reply Optimization

We increased the hello message interval to reduce the control packet overheads on the channel since nodes are less likely to be transmitting when a route request is initiated. The route waiting list for route replies was expanded to accommodate a greater number of entries, as for the hello message list. Furthermore, the route request and route reply SDU sizes were slightly increased, which also had a good impact on the PDR, as well as the throughput, as observed in preliminary experiments that are not reported in this study. As a result, we used the same value in all of the tests reported.

5.1.2. Route Request Search and TTL Optimization

To improve the protocol's efficiency in terms of delays, we reduced the ring buffer delay, the route request search TTL, and the route request interval timings while improving the routing table's entries to accommodate more routes.

5.1.3. BLE Mesh Network and AODV Layer Optimization

Furthermore, in order to reduce the likelihood of unwanted retransmissions, we reduced the channel utilization and probability of collision due to retransmissions, as well as the packet loss by lowering the packet's time-to-live in both the application and network layers after making the necessary changes in the mesh protocol's network layer's bt_ mesh_net_send and bt_ mesh_ net _recv functions. In addition, we improved the AODV protocol's rreq_send, rreq_ recv, and rrep_ send functions to reduce packet retransmission.

5.1.4. Application Layer Optimization

Because the application layer is critical for the compilation of the results, we improved the periodic updates function for printing status messages. Furthermore, we optimized the bt_ mesh_ cfg_ srv function used in the Zephyr RTOS to modify various parameters required for our experimentation, such as enabling the relay, TTL value, GATT proxy, and so on, to improve application performance in accordance with our experimental requirements.

Subsequently, Figure 7 depicts the summary of the acquired optimizations (obtained with the help of the aforementioned hardware and software) that resulted in better and more efficient results. To begin, this research concentrated on optimizing the application layer to achieve the desired output, as well as printing the messages to the PuTTY terminal. In addition, we enabled some of the capabilities of the configuration server model function and streamlined the periodic update function to obtain the desired results. In addition, as indicated in Figure 7, changes were made to the AODV layer functions. Following that, to address the issue of message retransmission, the mesh network layer functions such as bt_ mesh_ net_ send and bt_mesh_net_recv were enhanced. In addition, the controller configuration was altered, as shown in Figure 7, to ensure that the protocol ran smoothly as per the requirement.

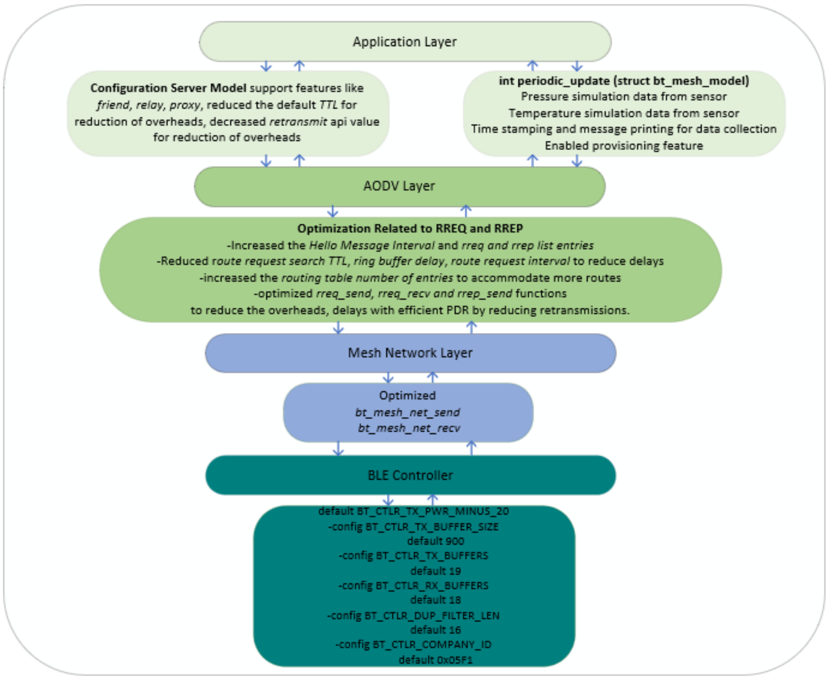
Application Layer
Configuration Server Model support features like *friend, relay, proxy*, reduced the default *TTL* for reduction of overheads, decreased *retransmit* api value for reduction of overheads
int periodic_update (struct bt_mesh_model)
Pressure simulation data from sensor
Temperature simulation data from sensor
Time stamping and message printing for data collection
Enabled provisioning feature
AODV Layer
Optimization Related to RREQ and RREP
-Increased the *Hello Message Interval* and *rreq and rrep list entries*
-Reduced *route request search TTL, ring buffer delay, route request interval* to reduce delays
-increased the *routing table number of entries* to accommodate more routes
-optimized *rreq_send, rreq_recv and rrep_send* functions
to reduce the overheads, delays with efficient PDR by reducing retransmissions.
Mesh Network Layer
Optimized
bt_mesh_net_send
bt_mesh_net_recv
BLE Controller
default BT_CTLR_TX_PWR_MINUS_20
-config BT_CTLR_TX_BUFFER_SIZE
default 900
-config BT_CTLR_TX_BUFFERS
default 19
-config BT_CTLR_RX_BUFFERS
default 18
-config BT_CTLR_DUP_FILTER_LEN
default 16
-config BT_CTLR_COMPANY_ID
default 0x05F1

Figure 7. Flowchart depicting the protocol optimizations.

5.1.5. Impact of Optimizations

In the light of above-mentioned optimizations, the proposed protocol exhibited practically constant route request and route reply to setup time and less overhead when compared to the old AODV and mesh protocols. The BLE mesh had very high overheads due to packet retransmissions, which the proposed protocol eliminated. We used the network's relay feature to control the number of packet retransmissions to support directed multihop transmission. In addition, for improved protocol efficiency, we increased the buffer size in the controller for the queuing of a few extra Tx and Rx PDUs that showed better results in terms of packet loss and PDR.

5.2. Performance Measurement Metrics Used for the Experiments

To understand the experimental results in detail, it is necessary to first explain the metrics that we used to assess the overall performance of our protocol.

Average End-to-End Delay

The term "average end-to-end delay" or "average one-way delay" refers to the amount of time it takes a packet to travel across a network from its source to its destination. It was calculated as:

$$D = 1/\mathit{Number\ of\ Packets\ Successfully\ Delivered} \sum_{i=1}^{n} (Tr_i - Ts_i) \tag{1}$$

where *D* = Average End-to-End Delay
i = Packet Identifier
Tr = Reception Time
Ts = Send Time
n = Number of Packets Successfully Delivered

Average Per-Hop Delay

When data are sent from a source to a destination, the delay caused by each hop is referred to as the hop delay. It was calculated as:

$$\mathit{Average\ Per\text{-}Hop\ Delay} = \mathit{Average\ End\text{-}to\text{-}End\ Delay/Number\ of\ Hops\ from\ the\ Destination\ Node} \tag{2}$$

PDR

This ratio is defined as the proportion of packets that arrive at their destination in relation to the number of packets that were originally sent to the destination (source to destination). The overall performance improved when the packet delivery ratios were high. It was calculated as:

$$PDR = \sum \text{Packets Received by All Destination Nodes} / \sum \text{Packets Sent by All Source Nodes} \tag{3}$$

RREQ-RREP Setup Time

This is the total time taken by the protocol from the time the route request is initiated until it reaches the destination and the destination sends the route reply message to the source and is received by the source. It was calculated as:

$$\mathit{RREQ\text{-}RREP\ Setup\ Time} = \mathit{t\_rreq\_received - t\_rrep\_sent} \tag{4}$$

Overhead

We calculated the overhead by counting the number of packets retransmitted on the mesh network layer level, by adjusting the prompt message and the counter.

5.3. Common Experiment Setup and Configurations

We managed to run the experiments with three different topologies: linear, multipath, and partial mesh multipath. Furthermore, for each of the three scenarios, we experimented seven times to obtain precise results with 100 packets transmitted from the source with the lowest transmit power set to make the testbed manageable. We had nine sources and a sink in Scenario 1 and a single source and a sink in Scenarios 2 and 3. Furthermore, Table 2 shows the general experiment setup and configurations for the experiments.

Table 2. Common experiment setup and configurations.

Experiment Parameters	Scenario 1	Scenario 2	Scenario 3
No of Experiments	7	7	7
Topology	Linear Topology with 10 Nodes	Multipath Topology with 6 Nodes	Multipath Topology with 10 Nodes
Link Speed	2 Mbps	2 Mbps	2 Mbps
Offered Data Rate	10 PDUs per Second	10 PDUs per Second	10 PDUs per Second
Packet Size	15 Byte	15 Byte	15 Byte
No. of Packets Transmitted from the Source	100	100	100
Number of Sources and Sinks	9 Sources, 1 Sink	1 Source, 1 Sink	1 Source, 1 Sink
Transmission Power	−40 dBm	−40 dBm	−40 dBm
Transmission Range	Approximately: 2–2.5 m	Approximately: 2–2.5 m	Approximately: 2–2.5 m
Relay Configuration	Enabled	Enabled	Enabled
Provisioning	Enabled but Given Hardcoded Network, Application, and Device Keys	Enabled but Given Hardcoded Network, Application and Device Keys	Enabled but Given Hardcoded Network, Application, and Device Keys
Packet Duplication Factor	3 Transmissions with a 20 ms interval	3 Transmissions with a 20 ms interval	3 Transmissions with a 20 ms interval

5.4. *Experimental Results: O-AODV Linear Topology with 10 Nodes*

These studies were carried out using the linear topology of ten nodes. There are nine sources transmitting simultaneously towards the sink. Hence, the links closer to the sink carried the packets generated by previous nodes from the previous hops, resulting in the link closest to the sink being the bottleneck link with the highest amount of traffic.

This section summarizes the experimental findings in terms of the end-to-end delays, average per-hop delays, route request to route reply setup time, overhead, and PDR.

According to the results shown in Figure 8, the end-to-end delay increased linearly as the number of hops increased, and O-AODV was proven efficient in comparison with the other two protocols.

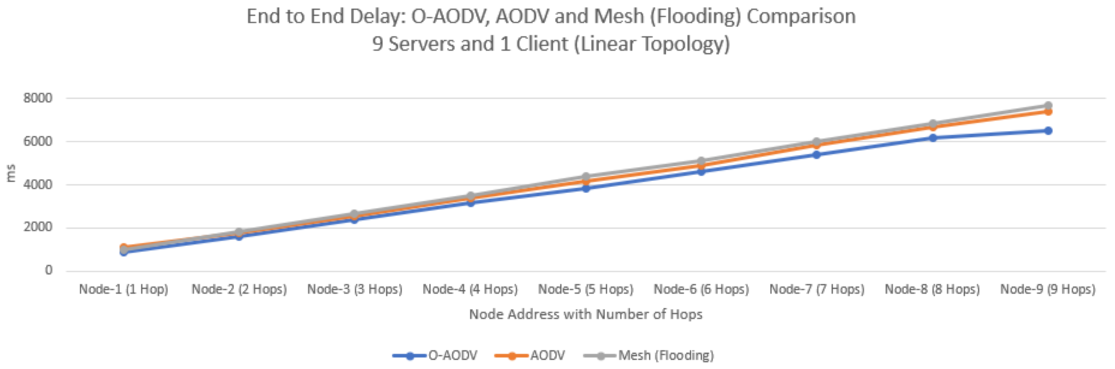

Figure 8. End-to-end, one-way delay: O-AODV, AODV, and mesh (flooding) comparison.

According to the results shown in Figure 9, the average per-hop delay for all three protocols was almost linear, with better O-AODV results than the other two protocols. We achieved an average per-hop delay of 840 ms compared to 1100 ms for AODV and 1000 ms for mesh (flooding) protocols.

As shown in the graph depicted in Figure 10, mesh (flooding) had a very high overhead of more than 80% compared to the O-AODV and AODV protocols. Furthermore, O-AODV incurred a much lower overhead by reducing message retransmissions and TTL values.

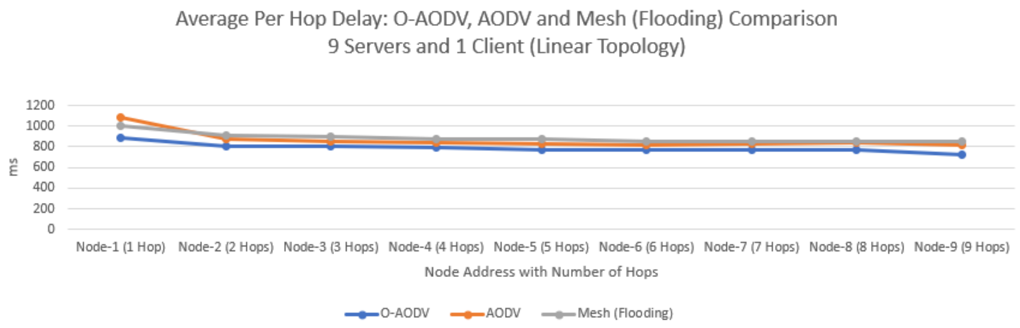

Figure 9. Average per-hop delay: O-AODV, AODV, and mesh (flooding) comparison.

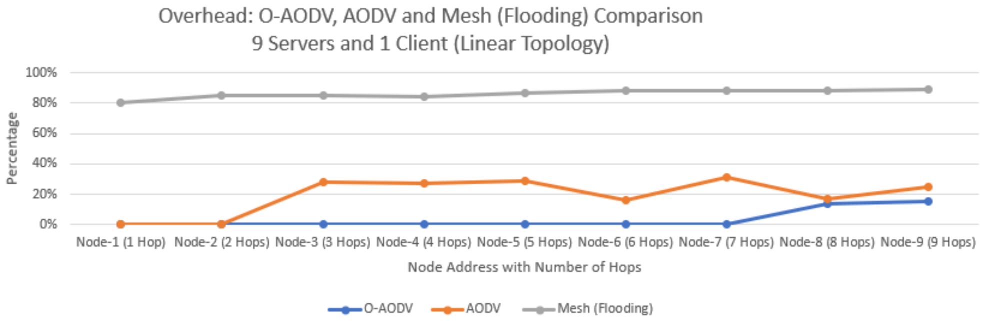

Figure 10. Overhead: O-AODV, AODV, and mesh (flooding) comparison.

As illustrated in Figure 11, both O-AODV and AODV showed a slight linear increase in the route request to route reply to setup time with an increase in the number of hops, with the O-AODV protocol taking approximately 16% less time.

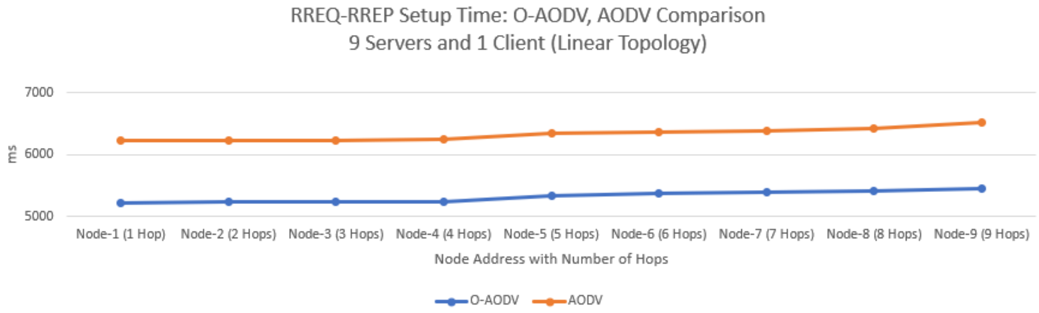

Figure 11. RREQ-RREP setup time: O-AODV and AODV comparison.

In Figure 12, O-AODV outperformed AODV in terms of the packet delivery ratio with 85% compared to 70%, but it was lower than the PDR for Mesh (Flooding), which was 95%.

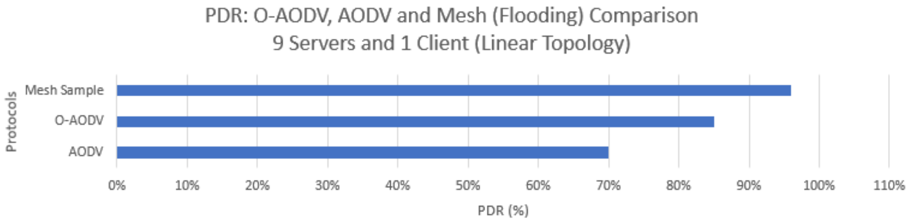

Figure 12. PDR: O-AODV, AODV, and mesh (flooding) comparison.

5.5. *Experimental Results: O-AODV Multipath Topology with Six Nodes*

In this section, we show the results of the experiments with the six-node AODV topology proposed by [6].

The single source transmits data to the sink via two hops through the mesh, which has four different possible routes.

The routing protocols in Figures 13 and 14 had less traffic load than the mesh (flooding) protocol, which had very heavy traffic loads due to uncontrolled traffic flow, as depicted in Figure 15. Furthermore, the routing protocols created less overhead traffic with route selection mechanisms, resulting in better results with O-AODV in light of the optimizations discussed in the preceding sections.

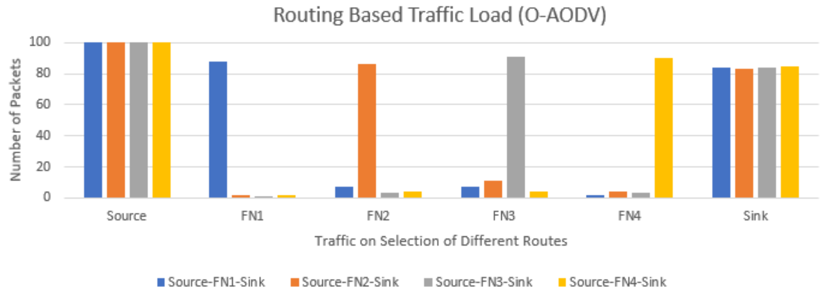

Figure 13. Traffic load: O-AODV.

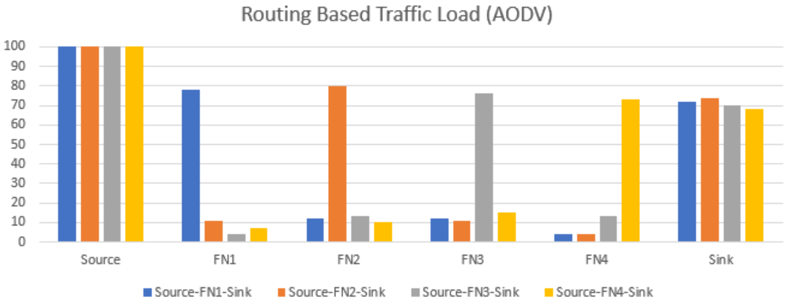

Figure 14. Traffic load: AODV.

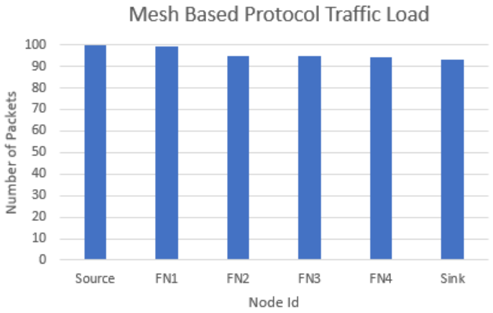

Figure 15. Traffic load mesh (flooding)-based protocol.

Furthermore, as shown in Figure 16, O-AODV had a better route request to route reply setup time of 5100 ms compared to 6100 ms for AODV.

Furthermore, the proposed optimized protocol had a lower average end-to-end delay of 4000 ms compared to 4500 ms for AODV and 6500 ms for the mesh (flooding) protocols, as shown in Figure 17. In addition, O-AODV achieved lower overheads of 20% compared to 40% for the AODV and 79% for mesh (flooding) protocols, as depicted in Figure 18. However, in terms of the PDR, O-AODV (84%) outperformed AODV (72%), but not at the mesh level (flooding) of 95%, as shown in Figure 19.

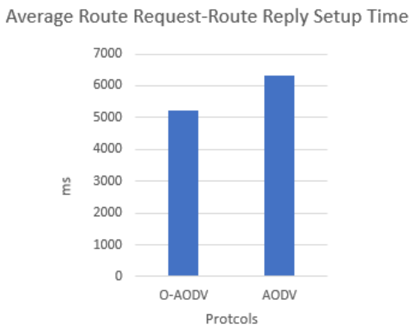

Figure 16. Average RREQ-RREP setup time.

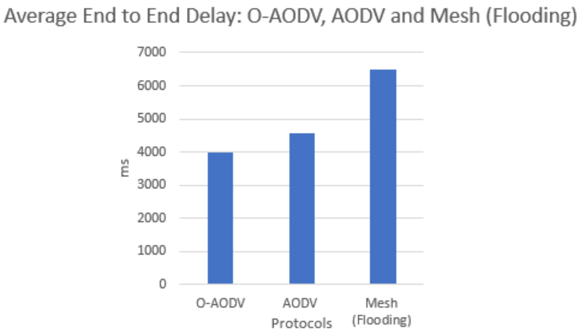

Figure 17. Average end-to-end delay: O-AODV, AODV, and mesh (flooding) comparison.

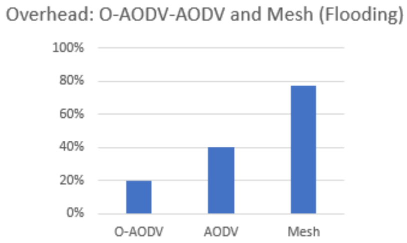

Figure 18. Overhead: O-AODV, AODV, and mesh (flooding).

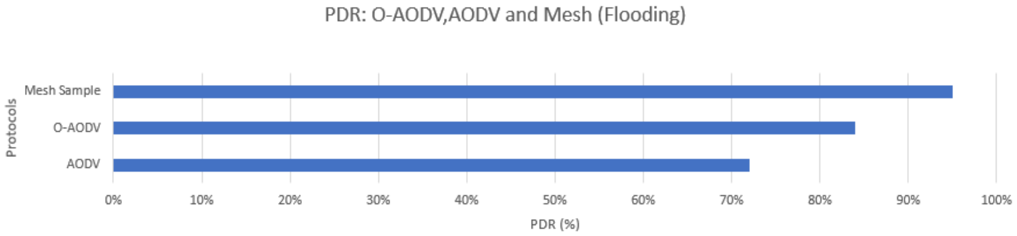

Figure 19. PDR: O-AODV, AODV, and mesh (flooding) comparison.

5.6. Experimental Results: O-AODV Multipath Topology with 10 Nodes

These experiments were conducted using the 10-node AODV partial mesh topology for further protocol testing. The findings are presented in this section.

The single source transmits data to the sink via four hops through the partial mesh, which has numerous possible routes.

Since there are various combinations of paths through the partial mesh, Figures 20 and 21 only present the results for the selected paths to illustrate the directed forwarding behavior of the O-AODV and AODV protocols.

The O-AODV and AODV routing protocols depicted in Figures 20 and 21 resulted in lower traffic loads with the ten-node complex/partial mesh scenario than the mesh (flooding) protocol, which resulted in extremely high traffic loads due to uncontrolled traffic flow, as depicted in Figure 22. Furthermore, due to the improvements discussed in the preceding sections, the proposed routing protocol caused less traffic with the routing mechanism, resulting in better results with O-AODV due to lesser packet retransmission overhead.

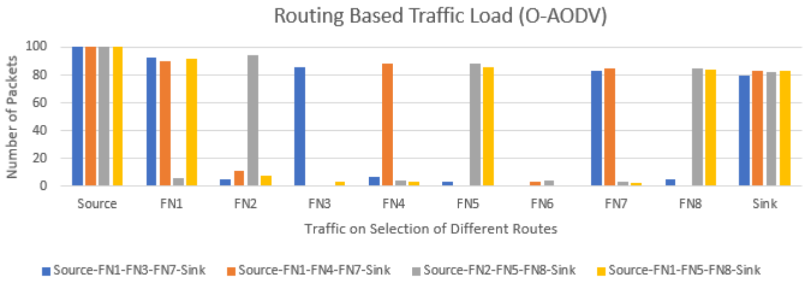

Figure 20. Traffic load: O-AODV.

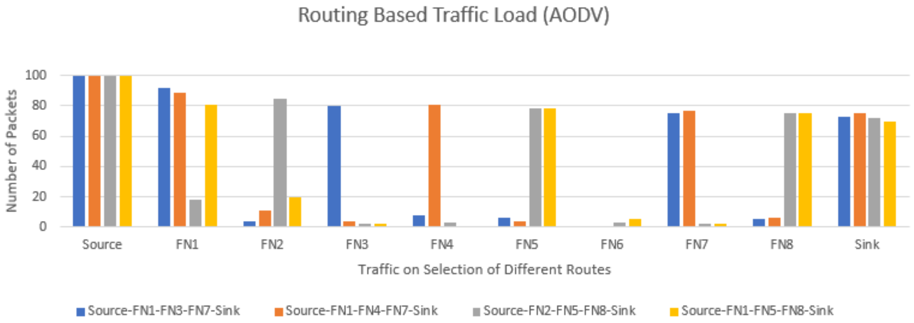

Figure 21. Traffic load: AODV.

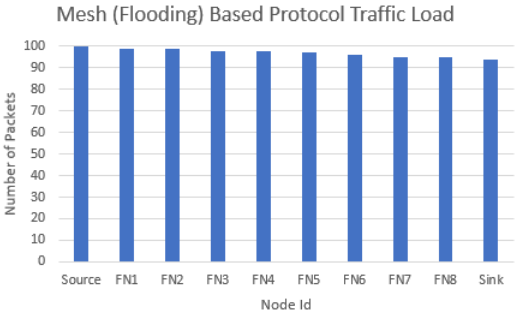

Figure 22. Traffic load: mesh (flooding)-based protocols.

In the partial mesh scenario with 10 nodes, O-AODV showed a lower average route request to route reply time of 5900 ms compared to 6800 ms for AODV, as illustrated in Figure 23, in comparison with its predecessor version due to the optimizations made to the mesh network and AODV protocol layers.

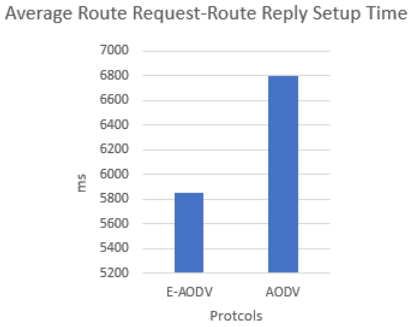

Figure 23. Average RREQ-RREP setup time.

Figure 24 shows that the O-AODV had a lower average end-to-end delay of 5300 ms compared to 6200 ms and 7200 ms for AODV and mesh (flooding), respectively, due to the controlled traffic load. Furthermore, O-AODV had lower overheads of 22% compared to 42% for the AODV and 80% for mesh (flooding) protocols due to less retransmission than AODV and mesh due to an improved routing mechanism, as shown in Figure 25. O-AODV also demonstrated its efficiency in terms of the PDR with 81% compared to AODV with 69%, but it was lower than mesh flooding, as shown in Figure 26, because mesh uses a flooding mechanism that favors the packet delivery ratio at the cost of much higher overheads.

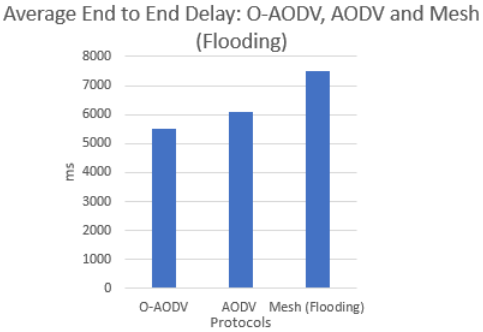

Figure 24. Average end-to-end delay: O-AODV, AODV, and mesh (flooding) comparison.

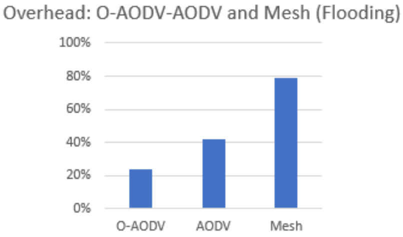

Figure 25. Overhead: O-AODV, AODV, and mesh (flooding).

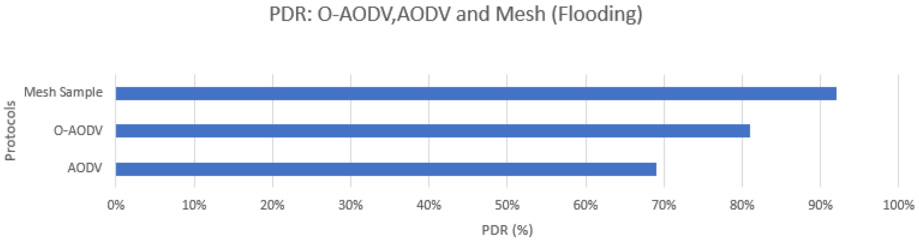

Figure 26. PDR: O-AODV, AODV, and mesh (flooding) comparison.

5.7. Experimental Result Findings

In the light of the above-mentioned protocol optimizations, we were able to obtain lower end-to-end delay and much lower overheads in comparison to mesh and the existing AODV protocol, as depicted in Figures 8, 10, 17, 18, 24 and 25 respectively.

In addition, the proposed protocol exhibited practically lower route requests and route reply setup time in comparison with AODV, as shown in Figures 11, 16 and 23.

In addition, the proposed protocol performed well in terms of efficiency by giving a better PDR than the AODV protocol for all three topology formats, as illustrated in Figures 12, 19 and 26.

Furthermore, it demonstrated that the traffic load for routing-based protocols was less than that for the mesh-based protocols, as shown in Figures 13–15 and 20–22.

5.8. Experiment Results, Discussion, and Research Contributions

The proposed O-AODV protocol performed better for various performance metrics, as shown in the improvements in the average end-to-end delay and overheads in comparison with AODV and flooding-based mesh. In addition, O-AODV also showed a much better RREQ-RREP setup time in comparison with AODV.

From the overhead data shown in all three scenarios, directed forwarding used by the routing protocols such as AODV and O-AODV was preferred over the flooding-based forwarding used in the BLE mesh standard since the high transmission overheads of up to 83% for flooding-based mesh would significantly reduce the operating lifetimes of battery-equipped sensor nodes.

The scalability of the O-AODV protocol with respect to the number of hops was evident from Scenario 1. Since average end-to-end delay increases with the number of hops, the average per-hop delay metric is useful for comparing the performance of all three protocols. In general, O-AODV achieved from 8% to over 20% lower average per-hop delay compared to the other protocols. In addition, O-AODV exhibited almost negligible overheads for up to seven hops, whereas AODV incurred up to a 25% overhead in comparison. Moreover, there was a slight increase in the overheads of eight and nine hops where traffic from multiple sources competed for the bottleneck link towards the sink, resulting in retransmissions, which allowing us to conclude that the optimum size of the multihop topology was seven hops or less. Furthermore, O-AODV incurred a lower RREP-RREQ setup time in comparison to AODV.

The lower average one-way delays and lower overheads for O-AODV contributed more significantly to the mesh topologies used in Scenarios 2 and 3. Since there were multiple possible paths between the source and sink, the number of duplicated forwarded packets in the flooding-based mesh protocol contributed significantly to the average end-to-end delay, requiring 6500 ms for the two-hop full mesh in Scenario 2 and 7500 ms for the four-hop partial mesh in Scenario 3. The increase was not linear with the number of hops due to the random nature of the flooding process in the mesh topology. In contrast, O-AODV achieved 4000 ms for the two-hop Scenario 2 and 5500 ms for the four-hop Scenario 3, which was lower than AODV with 4500 ms and 6000 ms in Scenario 2 and 3, respectively.

Subsequently, the average one-way delay for the mesh topology scenarios increased significantly as the number of hops through the network increased. The values in Figures 8, 17 and 24 clearly illustrate that the trend of end-to-end latency increased with the increasing number of nodes in the network. According to the mesh topology architecture, an increase in the number of full function nodes offering multihop routes led to significant network delay. Furthermore, with mesh, a greater number of full function nodes were involved in the mesh architecture, resulting in higher overall energy consumption.

The lower RREQ-RREP setup time for O-AODV in comparison to AODV was also significant. As the number of hops between the source and sink and the number of alternate paths within the mesh topology increased, the difference in the RREQ-RREP setup time of up to 1000 ms between O-AODV and AODV, as shown in Scenarios 2 and 3, would lead to much better responsiveness in terms of the networking joining time needed by new sensor nodes before they can start sending data to the sink.

Nonetheless, the improved efficiency of O-AODV in terms of the average one-way delay and overheads compared to the flooding-based mesh protocol had a tradeoff, in terms of the achieved Packet Delivery Ratio (PDR). For example, in Scenario 3, although O-AODV showed a higher PDR (82%) than AODV (69%), it was lower than the 92% achieved by the flooding-based mesh protocol. Nonetheless, that tradeoff may well be worthwhile given that the higher PDR of the flooding-based mesh was achieved only with much higher average one-way delays, whereas message delivery reliability can be improved via selective message retransmission in the application layer, which was expected to incur lower overheads and a lower increase in the end-to-end delays compared to the flooding-based mesh protocol.

In addition, for this research, we optimized the proposed protocol for use in scenarios involving static nodes since AODV was originally designed for mobile nodes.

6. Conclusions

This article presents the proposed O-AODV protocol for BLE mesh networks and its implementation on a testbed in comparison with the AODV and mesh protocols.

There were various challenges with the experimental approach that are worth noting. One of the issues we faced was the lack of a simulation environment, which prevented us from testing for a high number of nodes because with physical devices, it was difficult to scale the experiments to a large number of nodes. The transmission power of the nodes was also lowered to limit the testbed to a more manageable size of 20 m × 20 m. Finally, radio interference is an environmental factor that is difficult to eliminate in a physical testbed. This was reduced as much as possible by locating the tested in an area with minimal cochannel interference from WiFi access points and other wireless devices.

Coming to the result outcome of this research, in terms of the PDR, the proposed protocol (84%) performed better than AODV, but not to the level of the mesh protocol. Moreover, the developed protocol outperformed AODV and mesh in terms of overheads, end-to-end one-way delays, and average per-hop delays. Furthermore, the proposed protocol had a lower RREQ-RREP setup time than the previous protocol version.

The proposed optimization in O-AODV managed to reduce the delay to 5300 ms, which was less than the delays for AODV (6200 ms) and flooding mesh (7200 ms); while the overheads were also reduced to 22%, compared to the overheads for the AODV (42%) and flooding mesh (80%) protocols. Nonetheless, the PDR performance needs to be improved. In order to reduce the 10% difference in the PDR to be on par with the mesh protocol performance, enhancements such as multipath support for AODV to increase redundancy and ensure faster route recovery need to be investigated as the next step.

Furthermore, the reduced delays and overheads will undoubtedly benefit the hospital scenario, as hospitals deal with emergencies all of the time, and delays can have a significant impact on response time during emergencies. Furthermore, lowering the overheads will benefit the network by putting less load on it, as shown in the Figure 20. In addition, there is room for improvement in PDR to bring it on par with the mesh protocols for improved reliability, since applications requiring high reliability will need to retransmit data due to packet loss, decreasing the available traffic capacity and limiting the ability to scale the network to a larger number of nodes.

Author Contributions: M.R.G. and T.-C.W. developed and implemented the proposed protocol, as well as compiled the paper. M.R.G., T.-C.W., G.C.S., and A.R. all contributed to the selection of the paper, its organization, critical assessment, and manuscript drafting. The published version of the manuscript has been read and approved by all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AC	Advertising Channel
AFH	Adaptive Frequency Hopping
AODV	Ad hoc On-demand Distance Vector
ATT	Attribute Protocol
BLE	Bluetooth Low-Energy
DC	Data Channel
DSDV	Destination-Sequenced Distance Vector

DSR	Dynamic Source Control Routing
E2SR2	Energy-Efficient Secured Ring Routing
FM	FruityMesh
GAP	Generic Access Profile
GATT	Generic Attribute Profile
IETF	Internet Engineering Task Force
LL	Link Layer
LLP	Low-Energy Legacy Pairing
LESC	Low-Energy Secure Connection
L2CAP	Logical Link Control and Adaptation Protocol
MHTS	MultiHop Transfer Service
NDN	Named Data Networking
O-AODV	Optimized Ad hoc On-demand Distance Vector
OLSR	Optimized Link State Routing Protocol
PAN	Personal Area Network
PDR	Packet Delivery Ratio
PDU	Protocol Data Unit
SIG	Special Interest Group
TDMA	Time-Division Multiple Access
TTDD	Two-Tier Data Dissemination
WAHN	Wireless Ad Hoc Network

List of Symbols

The following symbols are used in this manuscript:

$\sum$	Summation sign
%	Percentage
$=$	Equals symbol
/	Division symbol
$-$	Subtraction (if used with a mathematical equation or formula)
D	Average end-to-end delay or average one-way delay
i	Packet identifier
$Mbps$	Mega bits per second
ms	Millisecond
n	Number of packets successfully delivered
Tr	Reception time
Ts	Send time

Appendix A. BLE Communication Profiles

Appendix A.1. Generic Access Profile

The Generic Access Profile (GAP) offers a standard structure for how Bluetooth Low-Energy (BLE) devices link. A BLE device can function as a broadcaster (advertises, but does not allow connections), an observer (sees advertisements, but does not initiate connections), a peripheral (advertises and accepts connections), or a central (sees advertisements and starts connections). BLE enables connectionless communications between broadcasters and observers through the use of advertisements (called beacons), as well as connection-oriented communications between peripheral and central devices. A BLE peripheral device, for example, broadcasts its presence at first, while the receiving device, a mobile phone acting as the central, establishes a connection with the peripheral to enable two-way communication. Following the formation of the connection, the central will operate as the master and the peripheral as the slave. To allow more complicated topologies, devices may also implement multiple roles.

Appendix A.2. Generic Attribute Profile

The Generic Attribute (GATT) profile specifies how data are transferred after the GAP has established a dedicated connection. It also specifies node roles, with one acting as the client and the other as the server.

Appendix A.3. Security Manager

The security manager specifies how devices are paired and keys are distributed. It provides secure communications and data transfer services to other layers.

Appendix A.4. Attribute Protocol

Clients and servers' roles are defined via the attribute protocol. A client sends requests to the server for reading and writing available attributes (data) that are stored on the server, while the server is in charge of storing the attributes and making them available to the client.

Appendix A.5. L2CAP

The L2CAP profile provides the upper layer with connection-oriented and connectionless data services, as well as multiplexing, segmentation, and reassembly capabilities.

Appendix B. Zephyr RTOS

Zephyr has a wide range of functions, including an extensive suite of kernel services (such as multithreading, interrupt, memory allocation, interthread synchronization, interthread data passing, power management), multiple scheduling algorithms, extreme modularity and configurability, support for cross-architecture, memory protection, compile-time resource definition, optimized device driver model, device tree support, Bluetooth Low-Energy 5.0, etc., [35].

Appendix B.1. Bluetooth Support

Zephyr OS is equipped with the Bluetooth Low-Energy controller (LE link layer) including BLE mesh and a Bluetooth BLE controller.

Furthermore, the OS supports the Generic Access Profile (GAP) with all LE roles, the Generic Attribute Profile (GATT), and the pairing with secure connections feature from Bluetooth 4.2. It has a smooth and clean HCI driver abstraction. Furthermore, a raw HCI interface can be utilized to operate Zephyr as a controller instead of using a full host stack that makes it extremely configurable.

Zephyr also offers full mesh support for protocol development, with features such as relay, friend node, Low-Power Node (LPN), and GATT proxy, as well as support for both provisioning bearers (PB-ADV and PB-GATT) and the ability to fit in devices with at least 16 k RAM.

There are three layers to the Zephyr Bluetooth stack such as the host (GAP, GATT, SM, ATT, L2CAP), controller (host control interface, link layer), and physical hardware. Furthermore, for application development, Zephyr comes with a number of BLE and mesh APIs.

Appendix B.2. BLE and BLE Mesh Security Features

In comparison to previous versions of Bluetooth, Version 5 introduced more advanced specifications to improve the functionalities of IoT-enabled devices with efficient and effective energy management [36]. The security dangers have increased as Bluetooth standards have improved, such as the coverage range (up to 200 m), as well as data throughput (2 Mbps). Similarly, with the enhanced range, attackers can gain access to the connection from a wider range of locations. Because of the aforementioned features, the BLE security mechanism differs from that of Bluetooth BR/EDR/HS. For security, one of two methods for pairing can be used for low-energy versions: Low-energy Legacy Pairing (LLP) or Low-Energy Secure Connection (LESC) (introduced in Version 4.2) [35].

Furthermore, the 4.2 and 5 specifications added a secure connection device pairing capability, as well as a new approach called numeric comparison, which has proven to be the best for BLE-based device authentication when compared to other methods.

Provisioning: Unprovisioned devices are those that are not connected to a BLE mesh network. If a device wants to join a mesh network, it should first go through the provisioning process [35]. The device will obtain the provision to enter the mesh network after provisioning and will be able to communicate with rest of the nodes in the network. In Zephyr, there are several API's to program a provisioning feature in the mesh-based applications [35].

References

1. Ghori, M.R.; Wan, T.C.; Sodhy, G.C. Bluetooth Low Energy 5 Mesh Based Hospital Communication Network (B5MBHCN). In *Advances in Cyber Security*; Springer: Singapore, 2020; Volume 1132, pp. 247–261. [CrossRef]
2. Ghori, M.R.; Wan, T.-C.; Sodhy, G.C. Bluetooth Low Energy Mesh Networks: Survey of Communication and Security Protocols. *Sensors* **2020**, *20*, 3590. [CrossRef] [PubMed]
3. Bluetooth. Available online: https://www.Bluetooth.com (accessed on 21 July 2021).
4. Perkins, C.; Das, S. *Ad Hoc On-Demand Distance Vector (AODV) Routing*; Tech. Memo.; University of California: Santa Barbara, CA, USA; University of Cincinnati: Cincinnati, OH, USA, 2003; Available online: https://tools.ietf.org/html/rfc3561 (accessed on 10 June 2021).
5. Johnson, D.; Hu, Y. *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*; Rice University: Houston, TX, USA, 2007; Available online: https://tools.ietf.org/html/rfc4728 (accessed on 5 June 2021).
6. Hussein, A.; Tarek, R.; Osama, H.; Fawzy, R.; Elsayed, K.; Taha, M. An AODV-Based Routing Scheme for Large-Scale Bluetooth Low-Energy Mesh Networks. In Proceedings of the 8th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC), Alexandria, Egypt, 14–15 December 2020; pp. 7–10. [CrossRef]
7. Zhang, J.; Sun, Z. Assessing multihop performance of reactive routing protocols in wireless sensor networks. In Proceedings of the IEEE International Conference on Communication Software and Networks (ICCSN), Beijing, China, 4–6 June 2016; pp. 444–449. [CrossRef]
8. Bhargava, N.; Bhargava, R.; Mathuria, M.; Kumawat, A. Performance Evaluation of Reactive and Proactive Routing Protocols over MANET. *Int. J. Comput. Appl.* **2013**, *73*, 3. [CrossRef]
9. Murillo, Y.; Reynders, B.; Chiumento, A.; Pollin, S. A Multiprotocol Low-Cost Automated Testbed for BLE Mesh. *IEEE Commun. Mag.* **2019**, *57*, 76–83. [CrossRef]
10. Clausen, T.; Jacquet, P. *Optimized Link State Routing Protocol (OLSR)*; Project Hipercom; INRIA: Paris, France, 2003; Available online: https://tools.ietf.org/html/rfc3626 (accessed on 5 June 2021).
11. Perkins, C.; Bhagwat, P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *ACM SIGCOMM Comput. Commun. Rev.* **1999**, *24*, 234–244. [CrossRef]
12. Jiang, M.; Li, J.; Hu, Y. *Cluster Based Routing Protocol (CBRP) Functional Specification*; National University of Singapore (NUS): Singapore, 1999.
13. Luo, H.; Ye, F.; Cheng, J.; Lu, S.; Zhang, L. TTDD: Two-Tier Data Dissemination in Large-Scale Wireless Sensor Networks. *Wirel. Netw.* **2005**, *11*, 161–175. [CrossRef]
14. Tunca, C.; Isik, S.; Donmez, M.Y.; Ersoy, C. Ring Routing: An Energy-Efficient Routing Protocol for Wireless Sensor Networks with a Mobile Sink. *IEEE Trans. Mob. Comput.* **2015**, *14*, 1947–1960. [CrossRef]
15. Bhushan, B.; Sahoo, G. E2SR2: An acknowledgement-based mobile sink routing protocol with rechargeable sensors for wireless sensor networks. *Wirel. Netw.* **2019**, *25*, 2697–2721. [CrossRef]
16. Mikhaylov, K.; Tervonen, J. Multihop data transfer service for Bluetooth Low Energy. In Proceedings of the 13th International Conference on ITS Telecommunications (ITST), Tampere, Finland, 5–7 November 2013; pp. 319–324. [CrossRef]
17. Martinez, C.; Eras, L.; Dominguez, F. The Smart Doorbell: A proof-of-concept Implementation of a Bluetooth Mesh Network. In Proceedings of the IEEE Third Ecuador Technical Chapters Meeting (ETCM), Cuenca, Ecuador, 15–19 October 2018; pp. 1–5. [CrossRef]
18. Balogh, A.; Imre, S.; Lendvai, K.; Szabo, S. Service Mediation in multihop Bluetooth Low Energy networks based on NDN approach. In Proceedings of the 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 16–18 September 2015; pp. 285–289. [CrossRef]
19. Guo, Z.; Harris, I.G.; Tsaur, L.; Chen, X. An on-demand scatternet formation and multihop routing protocol for BLE-based wireless sensor networks. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, LA, USA, 9–12 March 2015; pp. 1590–1595. [CrossRef]

20. Tanaka, K.; Murase, M.; Naito, K. Prototype implementation of BLE based automated data collection scheme in agricultural measurement system. In Proceedings of the 15th IEEE Annual Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018; pp. 1–2. [CrossRef]
21. Bardoutsos, A.; Filios, G.; Katsidimas, I.; Nikoletseas, S. Energy Efficient Algorithm for Multihop BLE Networks on Resource-Constrained Devices. In Proceedings of the 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini Island, Greece, 29–31 May 2019; pp. 400–407. [CrossRef]
22. Sirur, S.; Juturu, P.; Gupta, H.P.; Serikar, P.R.; Reddy, Y.K.; Barak, S.; Kim, B. A mesh network for mobile devices using Bluetooth low energy. In Proceedings of the IEEE Sensors, Busan, Korea, 1–4 November 2015. [CrossRef]
23. Ng, P.C.; She, J. A Novel Overlay Mesh with Bluetooth Low EnergyNetwork. In Proceedings of the WCNC, Marrakesh, Morocco, 15–18 April 2019; pp. 1–6. [CrossRef]
24. Dvinge, R.T.E.; Stalmach, A.; Nalpantidis, L. Connection-Based Bluetooth Mesh Network as a Low Energy Solution for Off-Grid Data Networks. In Proceedings of the 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, 13–15 May 2019; pp. 1–6. [CrossRef]
25. Li, R.; Li, X. Directional Multi-path Routing Algorithm Based on BLE Mesh. In Proceedings of the CSQRWC, Taiyuan, China, 18–21 July 2019; pp. 1–3. [CrossRef]
26. Leonardi, L.; Patti, G.; Lo Bello, L. Multi-Hop Real-Time Communications Over Bluetooth Low Energy Industrial Wireless Mesh Networks. *IEEE Access* **2018**, *6*, 26505–26519. [CrossRef]
27. Patti, G.; Leonardi, L.; Lo Bello, L. A Bluetooth Low Energy real-time protocol for Industrial Wireless mesh Networks. In Proceedings of the IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 4627–4632. [CrossRef]
28. Jung, C.; Kim, K.; Seo, J.; Silva, B.; Han, K. Topology Configuration and Multihop Routing Protocol for Bluetooth Low Energy Networks. *IEEE Access* **2017**, *5*, 9587–9598. [CrossRef]
29. Seymer, P.; Wijesekera, D.; Kan, C. Secure Outdoor Smart Parking Using Dual Mode Bluetooth Mesh Networks. In Proceedings of the VTC2019-Spring, Kuala Lumpur, Malaysia, 28 April–1 May 2019; pp. 1–7. [CrossRef]
30. Hansen, E.A.; Nielsen, M.H.; Serup, D.E.; Williams, R.J.; Madsen, T.K.; Abildgren, R. On Relay Selection Approaches in Bluetooth Mesh Networks. In Proceedings of the 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Moscow, Russia, 5–9 November 2018; pp. 1–5. [CrossRef]
31. Wang, S.; Chiang, K. BLE Tree Networks for Sensor Devices in Internet of Thing. In Proceedings of the DASC, PiCom, DataCom, CyberSciTech, Orlando, FL, USA, 6–10 November 2017; pp. 1304–1309. [CrossRef]
32. Chiumento, A.; Reynders, B.; Murillo, Y.; Pollin, S. Building a connected BLE mesh: A network inference study. In Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, 15–18 April 2018. [CrossRef]
33. Darroudi, S.M.; Gomez, C. Modeling the Connectivity of Data-Channel-Based Bluetooth Low Energy Mesh Networks. *IEEE Commun. Lett.* **2018**, *22*, 2124–2127. [CrossRef]
34. Renode. Available online: https://renode.io/ (accessed on 29 June 2021).
35. Zephyr Project. Available online: https://www.zephyrproject.org/ (accessed on 29 June 2021).
36. Silva, L.A.; Leithardt, V.R.Q.; Rolim, C.O.; González, G.V.; Geyer, C.F.R.; Silva, J.S. PRISER: Managing Notification in Multiples Devices with Data Privacy Support. *Sensors* **2019**, *19*, 3098. [CrossRef] [PubMed]